❖ NAME

Dalia Yehia Abdelaziz Ahmed Elwakeel.

❖ TITLE ABOUT

Simple and Efficient Pattern Matching Algorithms for Biological Sequences

❖ DR. SARA EL-METWALLY

❖ ABSTRACT

The phenomenal growth of biological data is an impetus to accelerate the discovery of solutions in a wide range of computational bioinformatics domains. Pattern matching is a very useful process in various stages of the computational pipelines. for example,Pattern matching, allows users to locate specific DNA subsequences in a database or DNA chain.Any trends in these ever-expanding biological datasets are changed over time. High-speed pattern matching algorithms are needed for faster searches. The current paper presents three pattern matching algorithms that have been designed specifically to speed up searches on massive DNA sequences. The proposed algorithms improve efficiency by using word processing (rather than the character processing presented in previous works) and also by checking the sequence for the least frequent word of the pattern. In terms of time cost, the experimental results show that the presented algorithms outperform the other simulated algorithms.

❖ *INTRODUCTION*

A text, sequence, or database is scanned to detect the positions of a pattern in the text in the pattern matching problem .This type of issue must be solved for many reasons sush as image, text and signal processing and chemistry.. The pattern matching problem is prevalent in various areas of computational bioinformatics, such as simple local alignment search, biomarker discovery, sequence alignment.. In these fields, it is necessary to recognise the locations of multiple patterns in databases, including those of amino acids and nucleotides. Gene analysis and DNA sequences may be used to investigate potential illness or abnormality diagnosis in biotechnology, forensics, pathology.. DNA sequence analysis may be used to compare a gene to related genes in the same or different species also to predict its function. In a different application, the functionality of a newly discovered DNA sequence can be predicted by comparing it to existing DNA sequences. This method has been used in a number of medical trials and applications. Despite the existence of some generalised and specialised DNA pattern matching algorithms in the literature, more powerful algorithms are required. Because of their high computing costs, many existing algorithms might not be well scalable for databases or massive DNA sequences. In comparison to approximate pattern matching, the current paper focuses on the actual pattern matching problem, which identifies all occurrences of a pattern in a text. it introduces three algorithms to address the shortcomings of previous works.The algorithms was split into two phases: preprocessing and matching. During preprocessingThe text's possible intervals for matching with the pattern are identified.These candidate intervals are called windows.The windows are then carefully scanned during the matching process to be aligned with the pattern. The less windows discovered during the preprocessing phase, the less time spent in the matching phase verifying the windows.The first suggested algorithm in the current analysis seeks the windows by considering both the first and last character of the pattern at the same time. Almost all processors nowadays have a computational length of 32 or 64 bits in each execution period. To put it another way, they can handle 4 or 8 bytes of data in a split second. As a result, 4 or 8 characters denoted by a word may be compared to 4 or 8 other characters at the same time. second algorithm for performing word-based comparisons in this paper. The word processing is done using The processor's computing power. This approach produces a new class of string-matching algorithms that outperform character-based algorithms. The current work reduces the number of detected windows and speeds up the comparisons by using this approach. As a result, the efficiency in terms of time cost increases.A third algorithm that focuses on the pattern word with the fewest text repetitions(the algorithm looks for a low-frequency pattern word in the code). By reducing the number of discovered windows, this technique improves the algorithm's performance.